

E-fólio Recurso

No e-fólio Recurso pretende-se implementar uma aplicação para codificar um ficheiro de texto, com um algoritmo muito particular, em que apenas a posição dos caracteres é alterada, mas não o seu valor¹: ordenar as linhas do ficheiro por tamanho; rodar os caracteres verticalmente N posições em cada coluna; rodar o resultado horizontalmente N posições em cada linha; repôr a ordem original das linhas. Ao descodificar, bastará utilizar a chave –N, tendo de se inverter a operação da rotação vertical com a horizontal.

Existem 4 alíneas, que devem ser realizadas sequencialmente, valendo cada alínea 1 valor. Os critérios de correção estão publicados no espaço da UC. Caso não consiga resolver todo o e-fólio, deve deixar por realizar as últimas alíneas. Deve entregar um único ficheiro c, com o seu número de estudante, independentemente das alíneas que realizou. No relatório deve indicar as alíneas realizadas, devendo o programa responder corretamente de acordo com a última alínea realizada (não necessita manter a interface das alíneas anteriores).

Não deve assumir nenhum valor máximo para o tamanho do ficheiro, e correspondentes número de linhas. Pode no entanto assumir que uma linha do ficheiro não tem mais de 255 caracteres, mas não deve alocar mais memória do que o necessário. Deve alocar memória dinamicamente, dependente do ficheiro e dados introduzidos pelo utilizador. Caso não deixe em aberto o número de linhas e respetivos tamanhos, utilize um valor máximo de 10.000 linhas, e para cada linha utilize 80 caracteres, sendo no entanto aplicável uma penalização de 50% nas alíneas C e D.

Deve entregar um relatório, descrevendo o trabalho realizado, os testes que fez, as opções que tomou, o qual não deve ultrapassar as 4 páginas nem conter o código fonte. Se realizou apenas parcialmente uma dada alínea, descreva o que fez nessa alínea no relatório. Não apresente um programa que não compile no TCC.

Alínea A) Ler ficheiro (1 valor)

Nesta alínea pretende-se que receba o nome de um ficheiro, e mostre o ficheiro. Considere que uma linha não tem mais de 255 caracteres.

Na primeira execução de exemplo não foi fornecido o nome do ficheiro, sendo indicado o modo de utilização. Na segunda execução o ficheiro foi lido e mostrado o seu conteúdo.

Execuções de exemplo:

```
C:\...>111111
Utilizacao: 111111 <ficheiro>
C:\...> 111111 teste.txt
ABCDEF
GHIJ
LMNOP
```

¹ O algoritmo solicitado tem apenas fins didáticos, não deve ser utilizado para codificar/descodificar ficheiros.

Alínea B) Ordenar linhas (1 valor)

Pretende-se nesta alínea que mostre as linhas do ficheiro, mas por ordem de tamanho. Em caso de empate, deve ser utilizada a ordem original do ficheiro.

Nas execução de exemplo pode-se ver o mesmo ficheiro utilizado na alínea anterior, em que as linhas são mostradas por ordem de tamanho.

Execuções de exemplo:

```
C:\...>111111 teste.txt
ABCDEF
LMNOP
GHIJ
```

Alínea C) Rotação horizontal (1 valor)

Pretende-se nesta alínea pretende-se que faça uma rotação de N caracteres em cada uma das linhas. O valor N deve ser dado nos parâmetros do programa.

Na primeira execução de exemplo, N=2, tendo sido utilizado o mesmo ficheiro que na alínea anterior. Pode-se ver que a letra A da primeira linha, deslocou-se duas posições para a direita. As duas últimas letras de cada linha, foram colocadas nas primeiras colunas da respectivas linhas. No caso com N=207, cada letra pode ter dado várias voltas na mesma linha, acabando por ficar 207 posições para a direita, considerando o número de colunas em cada linha.

Execução de exemplo:

```
C:\...>111111 teste.txt 2
EFABCD
OPLMN
IJGH

C:\...>111111 teste.txt 207
DEFABC
OPLMN
HIJG
```

Alínea D) Rotação vertical (1 valor)

Nesta alínea pretende-se que faça agora a rotação vertical nas colunas, e reponha a ordem original das linhas. Se N é positivo, a rotação das colunas deve ser realizada antes da rotação das linhas, se N é negativo, deve ocorrer a ordem contrária. No final deve ser reposta a ordem das linhas.

Na primeira execução de exemplo, pode-se ver o mesmo exemplo, agora com as colunas deslocadas, e reposicionadas as linhas. Vamos ver os estados que foram passados até à posição final:

Original	Ordenado	Colunas	Linhas	Final
ABCDEF GHIJ LMNOP	ABCDEF LMNOP GHIJ	LMNOEF GHIJP ABCD	EFLMNO JPGHI CDAB	EFLMNO CDAB JPGHI

Na segunda execução o conteúdo final do ficheiro codificado com 2 é colocado em teste2.txt, e pode ser confirmado que -2 retorna o ficheiro original. Na última execução, é dado um ficheiro maior, a resolução da alínea A deste e-fólio, e um valor de N elevado, sendo mostrado o início do ficheiro.

Execuções de exemplo:

```
C:\...>111111 teste.txt 2
EFLMNO
CDAB
JPGHI
```

```
C:\...>111111 teste2.txt -2
ABCDEF
GHIJ
LMNOP
```

```
C:\...>111111 111111A.c 21090
gmn\r",strr fic)!=NU{ wr cMAa[i- 210);a f
lunmostr, se oicheirodem da ma }r sne!f("r!ot, st);nas co
oiz5/õearlas linhadõ firo>"); por or no ecranior FIL prt f C:1
colunasal intopeioant op0!açchr) **argv) o o dheiropte a or,
} argstitdi rtr; m ("co da ords (N)
...
```